

A Stochastic Game Framework for Analyzing Computational Investment Strategies in Distributed Computing

Swapnil Dhamal, Walid Ben-Ameur, Tijani Chahed, Eitan Altman, Albert Sunny, and Sudheer Poojary

Abstract—We study a stochastic game framework with dynamic set of players, for modeling and analyzing their computational investment strategies in distributed computing. Players obtain a certain reward for solving the problem or for providing their computational resources, while incur a certain cost based on the invested time and computational power. We first study a scenario where the reward is offered for solving the problem, such as in blockchain mining. We show that, in Markov perfect equilibrium, players with cost parameters exceeding a certain threshold, do not invest; while those with cost parameters less than this threshold, invest maximal power. Here, players need not know the system state. We then consider a scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. Here, in Markov perfect equilibrium, only players with cost parameters in a relatively low range in a given state, invest. For the case where players are homogeneous, they invest proportionally to the ‘reward to cost’ ratio. For both the scenarios, we study the effects of players’ arrival and departure rates on their utilities using simulations and provide additional insights.



1 INTRODUCTION

DISTRIBUTED computing systems comprise computers which coordinate to solve large problems. In a classical sense, a distributed computing system could be viewed as several providers of computational power contributing to the power of a common central entity (e.g. in volunteer computing [1], [2]). The central entity could, in turn, use the combined power for either fulfilling its own computational needs or distribute it to the next level of requesters of power (e.g. by a computing service provider to its customers in a utility computing model). The center would decide the time for which the system is to be run, and hence the compensation or reward to be given out per unit time to the providers. This compensation or reward would be distributed among the providers based on their respective contributions. A provider incurs a certain cost per unit time for investing a certain amount of power. So, in the most natural setting where the reward per unit time is distributed to the providers in proportion to their contributed power, a higher power investment by a provider is likely to fetch it a higher reward while also increasing its incurred cost, thus resulting in a tradeoff.

Distributed computing has gained more popularity than ever owing to the advent of blockchain. Blockchain has found application in various fields [3], such as cryptocurrencies, smart contracts, security services, public services, Internet of Things, etc. Its functioning relies on a proof-of-work procedure, where miners (providers of computational power) collect block data consisting of a number of transactions, and repeatedly compute hashes on

inputs from a very large search space. A miner is rewarded for mining a block, if it finds one of the rare inputs that generates a hash value satisfying certain constraints, before the other miners. Given the cryptographic hash function, the best known method for finding such an input is randomized search. Since the proof-of-work procedure is computationally intensive, successful mining requires a miner to invest significant computational power, resulting in the miner incurring some cost. Once a block is mined, it is transmitted to all the miners. A miner’s objective is to maximize its utility based on the offered reward for mining a block before others, by strategizing on the amount of power to invest. There is a natural tradeoff: a higher investment increases a miner’s chance of solving the problem before others, while a lower investment reduces its incurred cost.

In this paper, we study the stochastic game where players (miners or providers of computational power) can arrive and depart during the mining of a block or during a run of volunteer computing. We consider two of the most common scenarios in distributed computing, namely, (1) in which the reward is offered for solving the problem (such as in blockchain mining) and (2) in which the reward is offered for contributing to the computational power of a common central entity (such as in volunteer computing).

1.1 Preliminaries

Stochastic Game. [4] It is a dynamic game with probabilistic transitions across different system states. Players’ payoffs and state transitions depend on the current state and players’ strategies. The game continues until it reaches a terminal state, if any. Stochastic games are thus a generalization of both Markov decision processes and repeated games.

Markov Perfect Equilibrium (MPE). MPE [5] is an adaptation of subgame perfect Nash equilibrium to stochastic games. An MPE strategy of a player is a policy function describing its strategy for each state, while ignoring history. Each player computes its best response strategy in each state by foreseeing the effects of its actions on the state transitions and the resulting utilities, and the strategies of other players. A player’s MPE policy is a best response to the other players’ MPE policies.

- Contact author: Swapnil Dhamal (swapnil.dhamal@gmail.com)
- Swapnil Dhamal is a postdoctoral researcher with Chalmers University of Technology, Sweden. A part of this work was done when he was a postdoctoral researcher with INRIA Sophia Antipolis-Méditerranée, France and Télécom SudParis, France. Walid Ben-Ameur and Tijani Chahed are professors with Télécom SudParis, France. Eitan Altman is a senior research scientist with INRIA Sophia Antipolis-Méditerranée, France. Albert Sunny is an assistant professor with Indian Institute of Technology, Palakkad, India. A part of this work was done when he was a postdoctoral researcher with INRIA Sophia Antipolis-Méditerranée, France. Sudheer Poojary is a senior lead engineer with Qualcomm India Pvt. Ltd. A part of this work was done when he was a postdoctoral researcher with Laboratoire Informatique d’Avignon, Université d’Avignon, France.

It is worth noting that, while game theoretic solution concepts such as MPE, Nash equilibrium, etc. may seem impractical owing to the common knowledge assumption, they provide a strategy profile which can be suggested to players (e.g. by a mediator) from which no player would unilaterally deviate. Alternatively, if players play the game repeatedly while observing each other's actions, they would likely settle at such a strategy profile.

1.2 Related Work

Stochastic games have been studied from theoretical perspective [6], [7], [8], [9], [10] as well as in applications such as computer networks [11], cognitive radio networks [12], wireless network virtualization [13], queuing systems [14], multiagent reinforcement learning [15], and complex living systems [16].

We enlist some of the important works on stochastic games. Altman and Shimkin [17] consider a processor-sharing system, where an arriving customer observes the current load on the shared system and chooses whether to join it or use a constant-cost alternative. Nahir et al. [18] study a similar setup, with the difference that customers consider using the system over a long time scale and for multiple jobs. Hassin and Haviv [19] propose a version of subgame perfect Nash equilibrium for games where players are identical; each player selects strategy based on its private information regarding the system state. Wang and Zhang [20] investigate Nash equilibrium in a queuing system, where reentering the system is a strategic decision. Hu and Wellman [21] use the framework of general-sum stochastic games to extend Q-learning to a noncooperative multiagent context. There exist works which develop algorithms for computing good, not necessarily optimal, strategies in a state-learning setting [22], [23].

Distributed systems have been studied from game theoretic perspective in the literature [24], [25]. Wei et al. [26] study a resource allocation game in a cloud-based network, with constraints on quality of service. Chun et al. [27] analyze the selfish caching game, where selfish server nodes incur either cost for replicating resources or cost for access to a remote replica. Grosu and Chronopoulos [28] propose a game theoretic framework for obtaining a user-optimal load balancing scheme in heterogeneous distributed systems.

Zheng and Xie [3] present a survey on blockchain. Sapirshstein et al. [29] study selfish mining attacks, where a miner postpones transmission of its mined blocks so as to prevent other miners from starting the mining of the next block immediately. Lewenberg et al. [30] study pooled mining, where miners form coalitions and share the obtained rewards, so as to reduce the variance of the reward received by each player. Xiong et al. [31] consider that miners can offload the mining process to an edge computing service provider. They study a Stackelberg game where the provider sets price for its services, and the miners determine the amount of services to request. Altman et al. [32] model the competition over several blockchains as a non-cooperative game, and hence show the existence of pure Nash equilibria using a congestion game approach. Kiayias et al. [33] consider a stochastic game, where each state corresponds to the mined blocks and the players who mined them; players strategize on which blocks to mine and when to transmit them.

In general, there exist game theoretic studies for distributed systems, as well as stochastic games for applications including blockchain mining (where a state, however, signifies the state of the chain of blocks). To the best of our knowledge, this work is the first to study a stochastic game framework for distributed

computing considering the set of players to be dynamic. We consider the most general case of heterogeneous players; the cases of homogeneous as well as multi-type players (which also have not been studied in the literature) are special cases of this study.

2 OUR MODEL

Consider a distributed computing system wherein agents provide their computational power to the system, and receive a certain reward for successfully solving a problem or for providing their computational resources. We first model the scenario where the reward is offered for solving the problem, such as in blockchain mining, and explain it in detail. We then model the scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. We hence point out the similarities and differences between the utility functions of the players in the two scenarios.

2.1 Scenario 1: Model

We present our model for blockchain mining, one of the most in-demand contemporary applications of the scenario where reward is offered for solving the problem. We conclude this subsection by showing that the utility function thus obtained, generalizes to other distributed computing applications belonging to this scenario.

Let r be the reward offered to a miner for successfully solving a problem, that is, for finding a solution before all the other miners.

Players. We consider that there are broadly two types of players (miners) in the system, namely, (a) strategic players who can arrive and depart while a problem is being solved (e.g., during the mining of a block) and can modulate the invested power based on the system state so as to maximize their expected reward and (b) fixed players who are constantly present in the system and invest a constant amount of power for large time durations (such as typical large mining firms). In blockchain mining, for instance, the universal set of players during the mining of a block consists of all those who are registered as miners at the time. In particular, we denote by \mathcal{U} , the set of strategic players during the mining of the block under consideration. We denote by ℓ , the constant amount of power invested by the fixed players throughout the mining of the block under consideration. We consider $\ell > 0$ (which is true in actual mining owing to mining firms); so the mining does not stall even if the set of strategic players is empty. Since the fixed players are constantly present in the system and invest a constant amount of power, we denote them as a single aggregate player k , who invests a constant power of ℓ irrespective of the system state.

Since it may not be feasible for a player to manually modulate its invested power as and when the system changes its state, we consider that the power to be invested is modulated by a pre-configured automated software running on the player's machine. The player can strategically determine the policy, that is, how much to invest if the system is in a given state.

We denote by cost parameter c_i , the cost incurred by player i for investing unit amount of power for unit time. We consider that players are not constrained by the cost they could incur. Instead, they aim to maximize their expected utilities (the expected reward they would obtain minus the expected cost they would incur henceforth), while forgetting the cost they have incurred thus far. That is, players are Markovian. In our work, we assume that the cost parameters of all the players are common knowledge. This could be integrated in a blockchain mining or volunteer computing interface where players can declare their cost parameters. This

information is then made available to the interfaces of all other players (that is, to the automated software running on the players' machines). In real world, it may not be practical to make the players' cost parameters a common knowledge, and furthermore, players may not reveal them truthfully. To account for such limitations, a mean field approach could be used by assuming homogeneous or multi-type players (which are special cases of our analysis). Furthermore, it is an interesting future direction to design incentives for the players to reveal their true costs.

Arrival and Departure of Players. For modeling the arrivals and departures of players, we consider a standard queueing setting. A player j , who is not in the system, arrives after time which is exponentially distributed with mean $1/\lambda_j$ (that is, the rate parameter is λ_j); this is in line with the Poission arrival process where the time for the first arrival is exponentially distributed with the rate parameter corresponding to the Poisson arrival. Further, the departure time of a player j , who is in the system, is exponentially distributed with rate parameter μ_j . The stochastic arrival of players is natural, like in most applications. Further, players would usually shut down their computers on a regular basis, or terminate the computationally demanding mining task (by closing the automated software) so as to run other critical tasks. Note that since players are Markovian, they do not account for how much computation they have invested thus far for mining the current block. Also, as we shall later see, the computation itself is memoryless, that is, the time required to find the solution does not depend on the time invested thus far. Owing to these two reasons, the players do not monitor block mining progress, and hence depart stochastically.

State Space. Due to the arrivals and departures of strategic players, we could view this as a continuous time multi-state process, where a state corresponds to the set of strategic players present in the system. So, if the set of strategic players in the system is S (which excludes the fixed players), we say that the system is in state S . So, we have $S \subseteq \mathcal{U}$ or equivalently, $S \in 2^{\mathcal{U}}$. In addition, we have $|\mathcal{U}| + 1$ absorbing states corresponding to the problem being solved by the respective player (one of the strategic players in \mathcal{U} or a fixed player). The players involved at any given time would influence each others' utilities, thus resulting in a game. The stochastic arrival and departure of players makes it a stochastic game. As we will see, there are also other stochastic events in addition to the arrivals and departures, and which depend on the players' strategies.

Players' Strategies. Let $\tau = 0$ denote the time when the mining of the current block begins. Let $x_i^{(S,\tau)}$ denote the strategy of player i (amount of power it decides to invest) at time τ when the system is in state S . Since players use a randomized search approach over a search space which is exponentially large as compared to the solution space, the time required to find the solution is independent of the search space explored thus far. That is, the search follows memoryless property. Also, note that a player has no incentive to change its strategy amidst a state owing to this memoryless property and if no other player changes its strategy. Hence in our analysis, we consider that no player changes its strategy within a state. So we have $x_i^{(S,\tau)} = x_i^{(S,\tau')}$ for any τ, τ' ; hence player i 's strategy could be written as a function of the state, that is, $x_i^{(S)}$. For a state S where $j \notin S$, we have $x_j^{(S)} = 0$ by convention. Let $\mathbf{x}^{(S)}$ denote the strategy profile of the players in state S . Let $\mathbf{x} = (\mathbf{x}^{(S)})_{S \subseteq \mathcal{U}}$ denote the policy profile.

TABLE 1
Notation

r	reward parameter
c_i	cost incurred by player i when it invests unit power for unit time
λ_i	arrival rate corresponding to player i
μ_i	departure rate corresponding to player i
\mathcal{U}	universal set of strategic players
ℓ	constant amount of power invested by the fixed players
k	aggregate player accounting for all the fixed players
S	set of strategic players currently present in the system
$x_i^{(S)}$	strategy of player i in state S
$\mathbf{x}^{(S)}$	strategy profile of players in state S
\mathbf{x}	policy profile
$\Gamma^{(S,\mathbf{x}^{(S)})}$	rate of problem getting solved in state S under strategy profile $\mathbf{x}^{(S)}$
$R_i^{(S,\mathbf{x})}$	expected utility of i computed in state S under policy profile \mathbf{x}

Rate of Problem Getting Solved. As explained earlier, the time required to find a solution in a large search space is independent of the search space explored thus far. We consider this time to be exponentially distributed to model its memoryless property (\because if a continuous random variable has the memoryless property over the set of reals, it is necessarily exponentially distributed). Let $\Gamma^{(S,\mathbf{x}^{(S)})}$ be the corresponding rate of problem getting solved in state S , when players' strategy profile is $\mathbf{x}^{(S)}$. Since the time required is independent of the search space explored thus far, the probability that a player finds a solution before others at time τ is proportional to its invested power at time τ .

Note that the time required for the problem to get solved is the minimum of the times required by the players to solve the problem. Now, the minimum of exponentially distributed random variables, is another exponentially distributed random variable with rate which is the sum of the rates corresponding to the original random variables. Furthermore, the probability of an original random variable being the minimum, is proportional to its rate. Let $P_j^{(S,\mathbf{x}^{(S)})}$ be the rate (corresponding to an exponentially distributed random variable) of player j solving the problem in state S , when the strategy profile is $\mathbf{x}^{(S)}$. So, we have $\Gamma^{(S,\mathbf{x}^{(S)})} = \sum_{j \in S \cup \{k\}} P_j^{(S,\mathbf{x}^{(S)})}$. Since the probability that player i solves the problem before the other players is proportional to its invested computational power at that time, we have that the rate of player i solving the problem is $P_i^{(S,\mathbf{x}^{(S)})} = \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$, and the rate of other players solving the problem is $Q_i^{(S,\mathbf{x}^{(S)})} = \sum_{j \in (S \setminus \{i\}) \cup \{k\}} P_j^{(S,\mathbf{x}^{(S)})} = \frac{\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell} \Gamma^{(S,\mathbf{x}^{(S)})}$.

The Continuous Time Markov Chain. Owing to the players being Markovian, when the system transits from state S to state S' , each player $j \in S \cap S'$ could be viewed as effectively reentering the system. So, the expected utility could be written in a recursive form, which we now derive. Table 1 presents the notation. The possible events that can occur in a state $S \in 2^{\mathcal{U}}$ are:

- 1) the problem gets solved by player i with rate $P_i^{(S,\mathbf{x}^{(S)})}$, thus terminating the game in the absorbing state where i gets a reward of r ;
- 2) the problem gets solved by one of the other players in $(S \setminus \{i\}) \cup \{k\}$ with rate $Q_i^{(S,\mathbf{x}^{(S)})}$, thus terminating the game in an absorbing state where player i gets no reward;

- 3) a new player $j \in \mathcal{U} \setminus S$ arrives and the system transits to state $S \cup \{j\}$ with rate λ_j ;
- 4) one of the players $j \in S$ departs and the system transits to state $S \setminus \{j\}$ with rate μ_j .

In what follows, we unambiguously write $j \in \mathcal{U} \setminus S$ as $j \notin S$, for brevity. Since $P_i^{(S, \mathbf{x}^{(S)})} + Q_i^{(S, \mathbf{x}^{(S)})} = \Gamma^{(S, \mathbf{x}^{(S)})}$, the sojourn time in state S is $(\Gamma^{(S, \mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j)^{-1}$. Let $D^{(S, \mathbf{x})} = \Gamma^{(S, \mathbf{x}^{(S)})} + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$. So, the expected cost incurred by player i while the system is in state S is $\frac{c_i x_i^{(S)}}{D^{(S, \mathbf{x})}}$.

Utility Function. The probability of an event occurring before any other event is equivalent to the corresponding exponentially distributed random variable being the minimum, which in turn, is proportional to its rate. So, player i 's expected utility as computed in state S is

$$R_i^{(S, \mathbf{x})} = \frac{\Gamma^{(S, \mathbf{x}^{(S)})} \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}}{D^{(S, \mathbf{x})}} \cdot r + \frac{\Gamma^{(S, \mathbf{x}^{(S)})} \frac{\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell}{\sum_{j \in S} x_j^{(S)} + \ell}}{D^{(S, \mathbf{x})}} \cdot 0 \\ + \sum_{j \notin S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} - \frac{c_i x_i^{(S)}}{D^{(S, \mathbf{x})}} \quad (1)$$

Note that we do not incorporate an explicit discounting factor with time. However, the utility of player i can be viewed as discounting the future owing to the possibility that the problem can get solved in a state S where $i \notin S$. Moreover, our analyses are easily generalizable if an explicit discounting factor is incorporated.

For distributed computing applications with a fixed objective such as finding a solution to a given problem, it is reasonable to assume that the rate of the problem getting solved is proportional to the total power invested by the providers of computation. We, hence, consider that $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$, where γ is the rate constant of proportionality determined by the problem being solved. Hence, player i 's expected utility as computed in state S is

$$R_i^{(S, \mathbf{x})} = (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S, \mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} \\ + \sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} \quad (2)$$

where $D^{(S, \mathbf{x})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right) + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$.

Other Applications of Scenario 1. We derived Expression (1) for the expected utility by considering that the probability of player i being the first to solve the problem is proportional to its invested power at the time, and hence obtains the reward r with this probability. Now, consider another type of system which aims to solve an NP-hard problem where players search for a solution, and the system rewards the players in proportion to their invested power when the problem gets solved. In this case, the first two terms of Expression (1) are replaced with the term $\frac{\Gamma^{(S, \mathbf{x}^{(S)})} \left(\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r \right)}{D^{(S, \mathbf{x})}}$. So, the mathematical form stays the same, and so when $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$, our analysis presented in Section 3 holds for this case too.

2.2 Scenario 2: Model

We now consider the scenario where the reward is offered for contributing to the computational power of a common central entity, such as in volunteer computing. Here, the reward offered per unit time is inversely proportional to the expected time for which the center decides to run the system. Considering that the time for which the center plans to run the system is exponentially distributed with rate parameter β , the reward offered per unit time is inversely proportional to $\frac{1}{\beta}$, and hence directly proportional to β . Hence, let the offered reward per unit time be $r\beta$, where r is the reward constant of proportionality. Furthermore, the reward given to a player is proportional to its computational investment. So, the revenue received by player i per unit time is $\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\beta$, and hence its net profit per unit time is $\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\beta - c_i x_i^{(S)}$. The sojourn time in state S , similar to the previous scenario, is $\frac{1}{D^{(S, \mathbf{x})}}$, where $D^{(S, \mathbf{x})} = \beta + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ (here, we have β instead of $\Gamma^{(S, \mathbf{x}^{(S)})}$). So, the net expected profit made by player i in state S before the system transits to another state, is $\frac{\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\beta - c_i x_i^{(S)}}{D^{(S, \mathbf{x})}}$.

Hence, player i 's expected utility as computed in state S is

$$R_i^{(S, \mathbf{x})} = \frac{\frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell} r\beta - c_i x_i^{(S)}}{D^{(S, \mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \cup \{j\}, \mathbf{x})} \\ + \sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot R_i^{(S \setminus \{j\}, \mathbf{x})} \quad (3)$$

Note that since $D^{(S, \mathbf{x})} = \beta + \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$ here, Expression (3) is obtainable from Expression (1), when $\Gamma^{(S, \mathbf{x}^{(S)})} = \beta$.

Other Variants of Scenario 2. We considered that the time for which the center decides to run the system is exponentially distributed with rate parameter β , where β is a constant. For theoretical interest, one could consider a generalization where the system may dynamically determine this parameter based on the set of players $S \cup \{k\}$ present in the system. Let such a rate parameter be given by $f(S)$. Since the fixed players and their invested power do not change, these could be encoded in $f(\cdot)$, thus making it a function of only the set of strategic players. The center could determine $f(S)$ based on the cost parameters of the players in set S , the past records of the investments of players in set S , etc. If the time for which the system is to run is independent of the set of players currently present in the system, we have the special case: $f(S) = \beta, \forall S$. It can be easily seen that the analysis presented in this paper (Section 4) goes through directly by replacing β with $f(S)$, since $\Gamma^{(S, \mathbf{x}^{(S)})} = f(S)$ is also independent of the players' investment strategies.

Further, note that if the rate parameter is not just dependent on the set of players present in the system but also proportional to their invested power, it could be written as $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$. This leads to the utility function being given by Equation (2) and hence its analysis is same as that of Scenario 1 (Section 3).

Convergence of Expected Utility

Note that Equation (1) encompasses both scenarios, where $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$ leads to Scenario 1, while $\Gamma^{(S, \mathbf{x}^{(S)})} = \beta$ leads to Scenario 2. We now show the convergence of this recursive equation, and hence derive a closed-form expression for utility function.

Let us define an ordering \mathcal{O} on sets which presents a one-to-one mapping from a set $S \subseteq \mathcal{U}$ to an integer between 1 and $2^{|\mathcal{U}|}$, both inclusive. Let $\mathbf{R}_i^{(\mathbf{x})}$ be the vector whose component $\mathcal{O}(S)$ is $R_i^{(S, \mathbf{x})}$. We now show that $\mathbf{R}_i^{(\mathbf{x})}$ computed using the recursive Equation (1), converges for any policy profile \mathbf{x} . Let $\mathbf{W}^{(\mathbf{x})}$ be the state transition matrix, among the states corresponding to the set of strategic players present in the system. In what follows, instead of writing $W^{(\mathbf{x})}(\mathcal{O}(S), \mathcal{O}(S'))$, we simply write $W^{(\mathbf{x})}(S, S')$ since it does not introduce any ambiguity. So, the elements of $\mathbf{W}^{(\mathbf{x})}$ are as follows:

$$\begin{aligned} \text{For } j \notin S : W_i^{(\mathbf{x})}(S, S \cup \{j\}) &= \frac{\lambda_j}{D^{(S, \mathbf{x})}} \\ \text{For } j \in S : W_i^{(\mathbf{x})}(S, S \setminus \{j\}) &= \frac{\mu_j}{D^{(S, \mathbf{x})}}, \end{aligned}$$

All other elements of $\mathbf{W}^{(\mathbf{x})}$ are 0.

Since $\ell > 0$, we have that $\Gamma^{(S, \mathbf{x}^{(S)})} > 0$. So, $D^{(S, \mathbf{x})} > \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j$. Hence, $\mathbf{W}_i^{(\mathbf{x})}$ is strictly substochastic (sum of the elements in each of its rows is less than 1).

Let $\mathbf{Z}_i^{(\mathbf{x})}$ be the vector whose component $\mathcal{O}(S)$ is $Z_i^{(S, \mathbf{x})}$, where

$$Z_i^{(S, \mathbf{x})} = \left(\frac{\Gamma^{(S, \mathbf{x}^{(S)})}}{\sum_{j \in S} x_j^{(S)} + \ell} r - c_i \right) \frac{x_i^{(S)}}{D^{(S, \mathbf{x})}},$$

Proposition 1. $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})}$.

Proof. Let $\mathbf{R}_{i(t)}^{(\mathbf{x})} = (R_{i(t)}^{(1, \mathbf{x})}, \dots, R_{i(t)}^{(2^{|\mathcal{U}|}, \mathbf{x})})^T$, where t is the iteration number and $(\cdot)^T$ stands for matrix transpose. The iteration for the value of $\mathbf{R}_{i(t)}^{(\mathbf{x})}$ starts at $t = 0$; we examine if it converges when $t \rightarrow \infty$. Now, the expression for the expected utility in all states can be written in matrix form and then solving the recursion, as

$$\begin{aligned} \mathbf{R}_{i(t)}^{(\mathbf{x})} &= \mathbf{W}^{(\mathbf{x})} \mathbf{R}_{i(t-1)}^{(\mathbf{x})} + \mathbf{Z}_i^{(\mathbf{x})} \\ &= (\mathbf{W}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} + \left(\sum_{\eta=0}^{t-1} (\mathbf{W}^{(\mathbf{x})})^\eta \right) \mathbf{Z}_i^{(\mathbf{x})} \end{aligned}$$

Now, since $\mathbf{W}^{(\mathbf{x})}$ is strictly substochastic, its spectral radius is less than 1. So when $t \rightarrow \infty$, we have $\lim_{t \rightarrow \infty} (\mathbf{W}^{(\mathbf{x})})^t = \mathbf{0}$. Since $\mathbf{R}_{i(0)}^{(\mathbf{x})}$ is a finite constant, we have $\lim_{t \rightarrow \infty} (\mathbf{W}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} = \mathbf{0}$. Further, $\lim_{t \rightarrow \infty} \sum_{\eta=0}^{t-1} (\mathbf{W}^{(\mathbf{x})})^\eta = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$ [34]. This implicitly means that $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})$ is invertible. Hence,

$$\begin{aligned} \lim_{t \rightarrow \infty} \mathbf{R}_{i(t)}^{(\mathbf{x})} &= \lim_{t \rightarrow \infty} (\mathbf{W}^{(\mathbf{x})})^t \mathbf{R}_{i(0)}^{(\mathbf{x})} + \left(\sum_{\eta=0}^{\infty} (\mathbf{W}^{(\mathbf{x})})^\eta \right) \mathbf{Z}_i^{(\mathbf{x})} \\ &= \mathbf{0} + (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})} \quad \square \end{aligned}$$

Owing to the requirement of deriving the inverse of $\mathbf{I} - \mathbf{W}^{(\mathbf{x})}$, it is clear that a general analysis of the concerned stochastic game when considering an arbitrary $\mathbf{W}^{(\mathbf{x})}$ is intractable. In this work, we consider two special scenarios that we motivated earlier in the context of distributed computing systems, for which we show that the analysis turns out to be tractable.

3 SCENARIO 1: ANALYSIS OF MPE

Let $\hat{R}_i^{(S, \mathbf{x})}$ be the equilibrium utility of player i in state S , that is, when i plays its best response strategy to the equilibrium strategies of the other players $j \in S \setminus \{i\}$ (while foreseeing effects of its actions on state transitions and resulting utilities). We can determine MPE similar to optimal policy in MDP (using policy-value iterations to reach a fixed point). Here, for maximizing $\hat{R}_i^{(S, \mathbf{x})}$, we could assume that we have optimized for other states and use those values to find an optimizing \mathbf{x} for maximizing $\hat{R}_i^{(S, \mathbf{x})}$. In our case, we have a closed form expression for vector $\mathbf{R}_i^{(\mathbf{x})}$ in terms of policy \mathbf{x} (Proposition 1); so we could effectively determine the fixed point directly.

A policy is said to be *proper* if from any initial state, the probability of reaching a terminal state is strictly positive. Consider the condition that, there exists at least one proper policy, and for any non-proper policy, there exists at least one state where the value function is negatively unbounded. It is known that, under this condition, the optimal value function is bounded, and it is the unique fixed point of the optimal Bellman operator [35]. Our model satisfies this condition, since there does not exist any non-proper policy as the probability of reaching a terminal state corresponding to the problem getting solved (either by player i or any other player including the fixed players) is strictly positive ($\because \Gamma^{(S, \mathbf{x}^{(S)})} > 0$).

Now, from Equation (2), the Bellman equations over states $S \in 2^{\mathcal{U}}$ for player i can be written as

$$\hat{R}_i^{(S, \mathbf{x})} = \max_{\mathbf{x}} \left\{ (\gamma r - c_i) \frac{x_i^{(S)}}{D^{(S, \mathbf{x})}} + \sum_{j \notin S} \frac{\lambda_j}{D^{(S, \mathbf{x})}} \cdot \hat{R}_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \frac{\mu_j}{D^{(S, \mathbf{x})}} \cdot \hat{R}_i^{(S \setminus \{j\}, \mathbf{x})} \right\}$$

We now derive some results, leading to the derivation of MPE.

Lemma 1. In Scenario 1, for any state S and policy profile \mathbf{x} , we have $R_i^{(S, \mathbf{x})} < r - \frac{c_i}{\gamma}$ if $\gamma r > c_i$, and $R_i^{(S, \mathbf{x})} > r - \frac{c_i}{\gamma}$ if $\gamma r < c_i$.

Proof. Let $V_i^{(S, \mathbf{x}^{(S)})}$ be the expected utility of player i in state S computed without considering the arrivals and departures of players ($\lambda_j = 0, \forall j \notin S$ and $\mu_j = 0, \forall j \in S$). So, we have

$$V_i^{(S, \mathbf{x}^{(S)})} = (\gamma r - c_i) \frac{x_i^{(S)}}{\gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)} = \left(r - \frac{c_i}{\gamma} \right) \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}$$

Let $\mathbf{V}_i^{(\mathbf{x})}$ be the vector whose component $\mathcal{O}(S)$ is $V_i^{(S, \mathbf{x}^{(S)})}$. Let $\mathbf{Z}_i^{(\mathbf{x})} = \mathbf{Y}^{(\mathbf{x})} \mathbf{V}_i^{(\mathbf{x})}$. Note that when $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$, we have that $\mathbf{Y}^{(\mathbf{x})}$ is a diagonal matrix, with diagonal elements $Y^{(\mathbf{x})}(S, S) = \frac{\gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)}{\gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)} = 1$. It can hence be seen that $\mathbf{W}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})}$ is a stochastic matrix (the sum of elements in each of its rows is 1).

Let $\mathbf{U}^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1}$, where $\mathbf{1}$ is the vector whose each element is 1. It is clear that all the elements of $\mathbf{U}^{(\mathbf{x})}$ are non-negative. We will now show that $\|\mathbf{U}^{(\mathbf{x})}\|_\infty \leq 1$, that is, the maximum element of the vector $\mathbf{U}^{(\mathbf{x})}$ is not more than 1. Let u_{S_0} be the element with the maximum value (one of the maximum, if there are multiple). Suppose $u_{S_0}^{(\mathbf{x})} = \|\mathbf{U}^{(\mathbf{x})}\|_\infty > 1$. So, we would have

$$\begin{aligned}
 \mathbf{U}^{(\mathbf{x})} &= (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1} \\
 \Rightarrow \mathbf{U}^{(\mathbf{x})} &= \mathbf{W}^{(\mathbf{x})} \mathbf{U}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})} \mathbf{1} \\
 \Rightarrow u_{S_0}^{(\mathbf{x})} &= \sum_{S \in 2^{\mathcal{U}}} u_S^{(\mathbf{x})} W^{(\mathbf{x})}(S_0, S) + Y^{(\mathbf{x})}(S_0, S_0) \\
 \Rightarrow u_{S_0}^{(\mathbf{x})} &< u_{S_0}^{(\mathbf{x})} \sum_{S \in 2^{\mathcal{U}}} W^{(\mathbf{x})}(S_0, S) + u_{S_0}^{(\mathbf{x})} Y^{(\mathbf{x})}(S_0, S_0) \\
 &\quad [\because \max_S u_S^{(\mathbf{x})} = u_{S_0}^{(\mathbf{x})} > 1] \\
 \Rightarrow \sum_{S \in 2^{\mathcal{U}}} W^{(\mathbf{x})}(S_0, S) + Y^{(\mathbf{x})}(S_0, S_0) &> 1
 \end{aligned}$$

However, this is a contradiction since $\mathbf{W}^{(\mathbf{x})} + \mathbf{Y}^{(\mathbf{x})}$ is a stochastic matrix. So, we have shown that $\|\mathbf{U}^{(\mathbf{x})}\|_{\infty} = \|(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{1}\|_{\infty} \leq 1$. That is, $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})}$ is either stochastic or substochastic.

From Proposition 1, $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})} \mathbf{V}_i^{(\mathbf{x})}$. Since $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Y}^{(\mathbf{x})}$ is stochastic or substochastic, $R_i^{(S, \mathbf{x})}$ for each S is a linear combination (with weights summing to less than or equal to 1) of $V_i^{(S, \mathbf{x}^{(S)})}$ over all $S \in 2^{\mathcal{U}}$.

For each S , $V_i^{(S, \mathbf{x}^{(S)})} = \left(r - \frac{c_i}{\gamma}\right) \frac{x_i^{(S)}}{\sum_{j \in S} x_j^{(S)} + \ell}$. So we have $V_i^{(S, \mathbf{x}^{(S)})} < r - \frac{c_i}{\gamma}$ if $\gamma r > c_i$, and $V_i^{(S, \mathbf{x}^{(S)})} > r - \frac{c_i}{\gamma}$ if $\gamma r < c_i$. Since $R_i^{(S, \mathbf{x})}$ for each S is a linear combination (with weights summing to less than or equal to 1) of $V_i^{(S, \mathbf{x}^{(S)})}$ over all $S \in 2^{\mathcal{U}}$, we have $R_i^{(S, \mathbf{x})} < r - \frac{c_i}{\gamma}$ if $\gamma r > c_i$, and $R_i^{(S, \mathbf{x})} > r - \frac{c_i}{\gamma}$ if $\gamma r < c_i$. \square

Lemma 2. In Scenario 1, $R_i^{(S, \mathbf{x})}$ is a monotone function of $x_i^{(S)}$.

Proof. We define the following for simplifying notation.

$$\begin{aligned}
 A_i^{(S, \mathbf{x})} &= \sum_{j \notin S} \lambda_j \hat{R}_i^{(S \cup \{j\}, \mathbf{x})} + \sum_{j \in S} \mu_j \hat{R}_i^{(S \setminus \{j\}, \mathbf{x})} \\
 E_i^{(S, \mathbf{x}^{(S)})} &= \sum_{j \notin S} \lambda_j + \sum_{j \in S} \mu_j + \gamma \left(\sum_{j \in S \setminus \{i\}} x_j^{(S)} + \ell \right)
 \end{aligned}$$

Hence, we can write

$$\begin{aligned}
 R_i^{(S, \mathbf{x})} &= \frac{A_i^{(S, \mathbf{x})} + (\gamma r - c_i) x_i^{(S)}}{E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}} \\
 \Rightarrow \frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}} &= \frac{(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma A_i^{(S, \mathbf{x})}}{\left(E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}\right)^2}
 \end{aligned}$$

The denominator is positive, while the numerator is a constant with respect to $x_i^{(S)}$, since $A_i^{(S, \mathbf{x})}$ and $E_i^{(S, \mathbf{x}^{(S)})}$ do not depend on $x_i^{(S)}$. So, $R_i^{(S, \mathbf{x})}$ is a monotone function of $x_i^{(S)}$, and whether it is increasing or decreasing, depends on the sign of the numerator: $(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma A_i^{(S, \mathbf{x})}$. \square

Proposition 2. In MPE for Scenario 1, a player i invests its maximal power if $\gamma r > c_i$, no power if $\gamma r < c_i$, and any amount of power if $\gamma r = c_i$.

Proof. Let $W^{(S, \mathbf{x})}$ be the row $\mathcal{O}(S)$ of $\mathbf{W}^{(\mathbf{x})}$. Note that $A_i^{(S, \mathbf{x})} = (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) W^{(S, \mathbf{x})} \hat{R}_i^{(\mathbf{x})}$. From the proof of Lemma 2, $\frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}}$ has same sign as $(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma A_i^{(S, \mathbf{x})}$, which can be written as

$$\begin{aligned}
 &(\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) W^{(S, \mathbf{x})} \hat{R}_i^{(\mathbf{x})} \\
 &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) (\hat{R}_i^{(S, \mathbf{x})} - Z_i^{(S, \mathbf{x})}) \\
 &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma \hat{R}_i^{(S, \mathbf{x})} (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) \\
 &\quad + \gamma \frac{(\gamma r - c_i) x_i^{(S)}}{E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}} (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) \\
 &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma \hat{R}_i^{(S, \mathbf{x})} (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) + \gamma (\gamma r - c_i) x_i^{(S)} \\
 &= (\gamma r - c_i) E_i^{(S, \mathbf{x}^{(S)})} - \gamma \hat{R}_i^{(S, \mathbf{x})} E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)} (\gamma r - c_i - \gamma \hat{R}_i^{(S, \mathbf{x})}) \\
 &= E_i^{(S, \mathbf{x}^{(S)})} (\gamma r - c_i - \gamma \hat{R}_i^{(S, \mathbf{x})}) + \gamma x_i^{(S)} (\gamma r - c_i - \gamma \hat{R}_i^{(S, \mathbf{x})}) \\
 &= (\gamma r - c_i - \gamma \hat{R}_i^{(S, \mathbf{x})}) (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}) \\
 &= \gamma \left(r - \frac{c_i}{\gamma} - \hat{R}_i^{(S, \mathbf{x})} \right) (E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)})
 \end{aligned}$$

Since $E_i^{(S, \mathbf{x}^{(S)})} + \gamma x_i^{(S)}$ is positive, and $(r - \frac{c_i}{\gamma} - \hat{R}_i^{(S, \mathbf{x})})$ has the same sign as $(\gamma r - c_i)$ from Lemma 1, we have that $\frac{dR_i^{(S, \mathbf{x})}}{dx_i^{(S)}}$ has the same sign as $(\gamma r - c_i)$. Also, note that if $\gamma r = c_i$, we have $R_i^{(S, \mathbf{x})} = 0, \forall S \in 2^{\mathcal{U}}$ from Proposition 1 when $\Gamma^{(S, \mathbf{x}^{(S)})} = \gamma \left(\sum_{j \in S} x_j^{(S)} + \ell \right)$.

So, in any state S , it is a dominant strategy for a player i to invest its maximal power if $\gamma r > c_i$, no power if $\gamma r < c_i$, and any amount of power if $\gamma r = c_i$. Since the maximal power of a player i would be bounded (let the bound be \bar{x}_i), it would invest \bar{x}_i if $\gamma r > c_i$. Hence, we have a consistent solution for the Bellman equations that a player i invests \bar{x}_i if $\gamma r > c_i$, 0 if $\gamma r < c_i$, and any amount of power in the range $[0, \bar{x}_i]$ if $\gamma r = c_i$. \square

Thus, the MPE strategy of a player follows a threshold policy, with a threshold on its cost parameter c_i (whether it is lower than γr) or alternatively, a threshold on the offered reward r (whether it is higher than $\frac{c_i}{\gamma}$). Note that though a player i invests maximal power when $\gamma r > c_i$, this is not inefficient since the power would be spent for less time as the problem would get solved faster. An intuition behind this result is that, when there are several miners in the system, the competition drives miners to invest heavily. On the other hand, when there are few miners in the system, miners invest heavily so that the problem gets solved faster (before arrival of more competition). Also, since the MPE strategies do not depend on S , the assumption of state knowledge can be relaxed.

We now provide an intuition for why the MPE strategies are independent of the arrival and departure rates. From Proposition 1, $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})}$. For $\gamma r > c_i$, when power $x_i^{(S)}$ increases, $\mathbf{Z}_i^{(\mathbf{x})}$ increases and $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$ decreases. But $\mathbf{R}_i^{(\mathbf{x})}$ increases with $x_i^{(S)}$ when $\gamma r > c_i$ (shown in the proof of Proposition 2), implying that the rate of increase of $\mathbf{Z}_i^{(\mathbf{x})}$ dominates the rate of decrease of $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$. So, the effect of $\mathbf{W}^{(\mathbf{x})}$ and hence state transitions is relatively weak, thus resulting in Markovian players playing strategies that are independent of the arrival and departure rates. Similar argument holds for $\gamma r \leq c_i$. It would be interesting to study scenarios where the rate of problem getting solved is a non-linear function of the players' invested powers. While a linear function is suited to most distributed computing applications, a non-linear function could possibly see $\mathbf{W}^{(\mathbf{x})}$ having a strong effect leading to MPE being dependent on the arrival and departure rates.

For analyzing the expected utility of a strategic player j , let us consider that the power available to it is very large,

say \bar{x}_j . Following our result on MPE, every player j satisfying $c_j < \gamma r$ would invest \bar{x}_j entirely. So, we have that $\gamma(\sum_{j \in S, c_j < \gamma r} \bar{x}_j + \ell)$ is very large, and hence $D^{(S, \mathbf{x})}$ (which now approximates to $\gamma(\sum_{j \in S, c_j < \gamma r} \bar{x}_j + \ell)$) is also very large. Since we know that $R_i^{(S \cup \{j\}, \mathbf{x})}$ and $R_i^{(S \setminus \{j\}, \mathbf{x})}$ are bounded by a small quantity from Lemma 1, the limit of the expected utility $R_i^{(S, \mathbf{x})}$ computed in any state S (from Equation (2)) is $\frac{\bar{x}_i}{\sum_{j \in S, c_j < \gamma r} \bar{x}_j + \ell} r - \frac{c_i \bar{x}_i}{\gamma(\sum_{j \in S, c_j < \gamma r} \bar{x}_j + \ell)}$. To get further insight into this, say ℓ is considered insignificant, that is, the computation is dominated by strategic players. Furthermore, say for every strategic player i , $c_i < \gamma r$, and let the very large amount of power available to these players be the same ($\bar{x}_i = \bar{x}_j, \forall i, j \in \mathcal{U}$). Thus, the limit of the expected utility $R_i^{(S, \mathbf{x})}$ computed in any state S simplifies to $\frac{r}{|\hat{S}|} - \frac{c_i}{\gamma|\hat{S}|}$. This is intuitive, since if $\bar{x}_i = \bar{x}_j, \forall i, j \in \mathcal{U}$, the reward would be won by the players with equal probability (hence the term $\frac{r}{|\hat{S}|}$), and the cost is reduced owing to the reduced time due to the combined rate of the problem getting solved (hence the term $\frac{c_i}{\gamma|\hat{S}|}$).

4 SCENARIO 2: ANALYSIS OF MPE

Proposition 3. *In MPE for Scenario 2, a player i invests*

$$x_i^{(S)} = \max \left\{ \psi^{(S)} \left(1 - \frac{c_i \psi^{(S)}}{r\beta} \right), 0 \right\}$$

where $\psi^{(S)} = r\beta \frac{|\hat{S}| - 1 + \sqrt{(|\hat{S}| - 1)^2 + \frac{4\ell}{r\beta} \sum_{j \in \hat{S}} c_j}}{2 \sum_{j \in \hat{S}} c_j}$. Here, \hat{S} is the maximal set of players $j \in S$ which collectively satisfy the constraints $c_j < \frac{r\beta}{\psi^{(S)}}$.

Proof. Recall that since $\mathbf{W}^{(\mathbf{x})}$ is a strictly substochastic matrix, $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} = \lim_{t \rightarrow \infty} \sum_{\eta=0}^{t-1} (\mathbf{W}^{(\mathbf{x})})^\eta$. Since all the elements of $\mathbf{W}^{(\mathbf{x})}$ are non-negative, all the elements of $(\mathbf{W}^{(\mathbf{x})})^\eta$ also are non-negative for any natural number η , and hence all the elements of $(\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1}$ are non-negative. Also, since $\mathbf{R}_i^{(\mathbf{x})} = (\mathbf{I} - \mathbf{W}^{(\mathbf{x})})^{-1} \mathbf{Z}_i^{(\mathbf{x})}$ (Proposition 1) and since $\mathbf{W}^{(\mathbf{x})}$ is independent of $x_i^{(S)}$ in this scenario, maximizing the components of $\mathbf{Z}_i^{(\mathbf{x})}$ (namely, $Z_i^{(S, \mathbf{x})}$) individually with respect to $x_i^{(S)}$ would essentially maximize all the elements of $\mathbf{R}_i^{(\mathbf{x})}$. Now, since $\Gamma^{(S, \mathbf{x}^{(S)})} = \beta$ in this scenario, we have

$$Z_i^{(S, \mathbf{x})} = \left(\frac{\beta}{\sum_{j \in S} x_j^{(S)} + \ell} r - c_i \right) \frac{x_i^{(S)}}{D^{(S, \mathbf{x})}}$$

As $D^{(S, \mathbf{x})}$ is independent of $x_i^{(S)}$ in this scenario, it can be shown that $Z_i^{(S, \mathbf{x})}$ is a concave function with respect to $x_i^{(S)}$ (the second derivative turns out to be $\frac{-2r\ell\beta}{(\sum_{j \in S} x_j^{(S)} + \ell)^3 D^{(S, \mathbf{x})}}$). The first order condition $\frac{dZ_i^{(S, \mathbf{x})}}{dx_i^{(S)}} = 0$ gives

$$x_i^{(S)} = \left(\sum_{j \in S} x_j^{(S)} + \ell \right) \left(1 - \frac{c_i}{r\beta} \left(\sum_{j \in S} x_j^{(S)} + \ell \right) \right)$$

Let $\psi^{(S)} = \sum_{j \in S} x_j^{(S)} + \ell$. As $x_i^{(S)}$ is non-negative, we have

$$x_i^{(S)} = \max \left\{ \psi^{(S)} \left(1 - \frac{\psi^{(S)}}{r\beta} c_i \right), 0 \right\} \quad (4)$$

Let $\hat{S} = \{j \in S : x_j^{(S)} > 0\}$. We later show how to determine set \hat{S} . Summing the above over all players in S and then adding ℓ on both sides, we get

$$\sum_{j \in S} x_j^{(S)} + \ell = \psi^{(S)} \left(|\hat{S}| - \frac{\psi^{(S)}}{r\beta} \sum_{j \in \hat{S}} c_j \right) + \ell$$

Substituting $\sum_{j \in S} x_j^{(S)} + \ell$ as $\psi^{(S)}$, we get

$$\frac{1}{r\beta} \sum_{j \in \hat{S}} c_j \left(\psi^{(S)} \right)^2 - (|\hat{S}| - 1) \psi^{(S)} - \ell = 0$$

Solving this equation for positive value of $\psi^{(S)}$, we get

$$\psi^{(S)} = r\beta \frac{|\hat{S}| - 1 + \sqrt{(|\hat{S}| - 1)^2 + \frac{4\ell}{r\beta} \sum_{j \in \hat{S}} c_j}}{2 \sum_{j \in \hat{S}} c_j}$$

Substituting this expression for $\psi^{(S)}$ in Equation (4) gives the MPE strategy of player i . \square

So, $x_i^{(S)} > 0$ iff $c_i < \frac{2 \sum_{j \in \hat{S}} c_j}{|\hat{S}| - 1 + \sqrt{(|\hat{S}| - 1)^2 + \frac{4\ell}{r\beta} \sum_{j \in \hat{S}} c_j}}$. That is,

only players with cost parameters in a relatively low range in a given state, invest. The constraint implies that if player i invests, then player j with $c_j < c_i$ also invests. So, there exists a threshold player \hat{i} such that any player j with $c_j > c_i$ would not invest. Hence, set \hat{S} can be constructed iteratively (initiating from an empty set) by adding players j from set $S \setminus \hat{S}$ one at a time, in ascending order of c_j , until the above constraint is violated for the cost parameter of the newly added player.

To get a better understanding of this result, if the power ℓ invested by fixed players is considered insignificant, we have $\psi^{(S)} = r\beta \frac{|\hat{S}| - 1}{\sum_{j \in \hat{S}} c_j}$ and the condition for $x_i^{(S)} > 0$ simplifies to $c_i < \frac{\sum_{j \in \hat{S}} c_j}{|\hat{S}| - 1}$.

Furthermore, if the strategic players are homogeneous ($c_i = c_j, \forall i, j \in \mathcal{U}$), the cost constraint is satisfied for all players in S (since $c < \frac{|\hat{S}|c}{|\hat{S}| - 1}$) and so, all the strategic players invest $\frac{r\beta}{c} \left(\frac{|\hat{S}| - 1}{|\hat{S}|^2} \right)$. That is, if the computation is dominated by strategic players which are homogeneous, they would invest proportionally to the ‘reward to cost parameter’ ratio in MPE.

Since the transition probabilities, and hence $\mathbf{W}^{(\mathbf{x})}$, are constant w.r.t. players’ strategies in this scenario, a player’s MPE utility computed in state S ($R_i^{(S, \mathbf{x})}$) is a linear combination (with constant non-negative weights) of its utilities over all states computed without accounting for state transitions. Hence, the MPE strategies are independent of the arrival and departure rates.

Note that while the decision regarding whether or not to invest was independent of the cost parameters of the other players in the system in Scenario 1, this decision highly depends on the cost parameters of other players in Scenario 2.

5 SIMULATION STUDY

Throughout the paper, we determined MPE strategies, which we observed to be independent of players’ arrival and departure rates. However, it is clear from Equations (1), (2), (3) and Proposition 1 that the players’ utilities would indeed depend on these rates. We now study the effects of these rates on the utilities in MPE using simulations. In order to reliably obtain an accurate relation between the arrival/departure rates and the expected utilities of the players, we consider that the computation is dominated by the strategic players (that is, the power invested by the fixed

players is insignificant: $\ell \rightarrow 0$) and the strategic players are homogeneous (their arrival/departure rates and their cost parameters are the same). Let λ, μ, c denote the common arrival rate, departure rate, and cost parameter, respectively. Note that if the strategic players are considered homogeneous, the players' sets (states) can be mapped to their cardinalities. We observe how the expected utility of a player changes as a function of the number of other players present in the system, for different arrival/departure rates. In our simulations, we consider the following values: $r = 10^5, \gamma = \beta = 0.1, |\mathcal{U}| = 10^4, c = 0.003$ (a justification of these values is provided in Appendix).

Statewise Nash Equilibrium. For a comparative study, we also look at the equilibrium strategy profile of a given set of players S , when there are no arrivals and departures ($\lambda_j = 0, \forall j \notin S$ and $\mu_j = 0, \forall j \in S$). We call this, *statewise* Nash equilibrium (SNE) in state S . Since the MPE strategies of the players are independent of the arrival and departure rates, a player's SNE strategy in a state is same as its MPE strategy corresponding to that state. Note, however, that the expected utilities in SNE would be different from those in MPE, since the expected utilities highly depend on the arrival and departure rates (Equations (1), (2), (3) and Proposition 1). Also, since SNE does not account for change of the set of players present in the system, the expected utilities in SNE for different values on X-axis in the plots are computed independently of each other.

5.1 Simulation Results

In Figures 1 and 2, the plots for expected utility largely follow near-linear curve (of negative slope) on log-log scale, with respect to the number of players in the system. That is, they nearly follow power law, which means that scaling the number of players by a constant factor would lead to proportionate scaling of expected utility.

Scenario 1. Figure 1 presents plots for expected utilities with MPE policy for various values of λ and μ , and compares them with expected utilities in SNE. Following are some insights:

- As seen at the end of Section 3, if the mining is dominated by strategic players which are homogeneous, the expected utilities in MPE are bounded by $\frac{r}{|S|} - \frac{c}{\gamma|S|}$. It can be similarly shown that the limit of the players' expected utilities in SNE is $\frac{r}{|S|} - \frac{c}{\gamma|S|}$ (this can be seen by substituting in Equation (2): $\lambda_j = 0 \forall j \notin S, \mu_j = 0, c_j = c, x_j^{(S)} \rightarrow \infty, \forall j \in S$, and $\ell \rightarrow 0$). Owing to this, the expected utilities in MPE are bounded by the expected utilities in SNE, which is reflected in Figure 1.
- In Scenario 1, a higher λ results in a higher likelihood of the system having more players, which results in a higher rate of the problem getting solved as well as more competition. This, in turn, reduces the time spent in the system as well as the probability of winning for each player, which hence reduces the cost incurred as well as the expected reward. Figure 1(a) suggests that, as λ changes, the change in cost incurred balances with the change in expected reward, since the change in expected utility is insignificant.
- For a given μ , if the number of players changes, there is a balanced tradeoff between the cost and the expected reward as above; so the change in expected utility is insignificant. But a higher μ results in a higher probability of player i departing from the system and staying out when the problem gets solved, thus lowering its expected utility (Figure 1(b)).

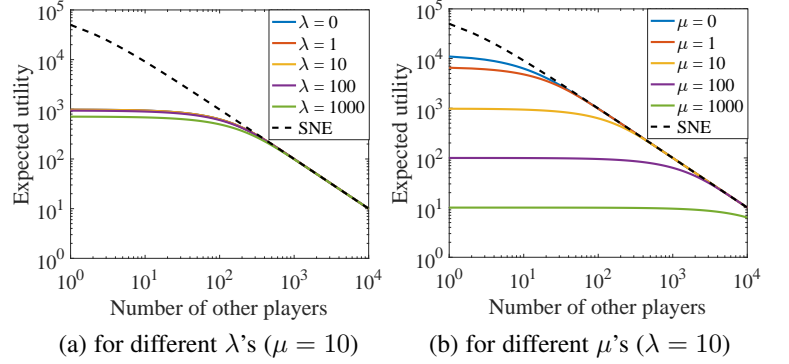


Fig. 1. Expected utility of a player in Scenario 1

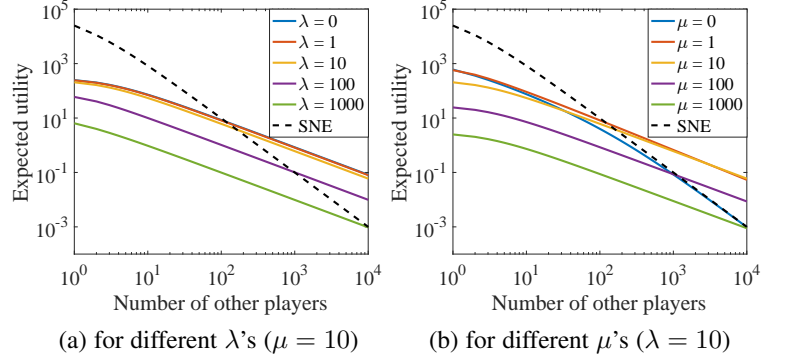


Fig. 2. Expected utility of a player in Scenario 2

Scenario 2. Since a player's SNE strategy in a state is same as its MPE strategy corresponding to that state, a player's SNE strategy is to invest $\frac{r\beta}{c} \left(\frac{|S|-1}{|S|^2} \right)$ in state S (as explained at the end of Section 4 when computation is dominated by strategic players that are homogeneous). Furthermore, in SNE, the expected utility of each player can be shown to be $\frac{r}{|S|^2}$ in state S (this can be seen by substituting in Equation (3): $\lambda_j = 0 \forall j \notin S, \mu_j = 0, c_j = c, x_j^{(S)} = \frac{r\beta}{c} \left(\frac{|S|-1}{|S|^2} \right), \forall j \in S$, and $\ell \rightarrow 0$). Figure 2 presents the plots for expected utilities with the analyzed MPE policy for different values of λ and μ , and compares them against SNE. Following are some insights:

- An increase in the number of players increases competition for the offered reward and hence reduces the reward per unit time received by each player, with no balancing factor (unlike in Scenario 1); so the expected utility decreases.
- For higher λ , there is higher likelihood of system having more players, thus resulting in lower expected utility owing to aforementioned reason. Also, from Figure 2(a), if λ is not very high, an increase in μ is likely to reduce the competition to the extent that the expected MPE utility when the number of players in the system is large, can exceed the corresponding SNE utility ($\frac{r}{|S|^2}$, which would be very low when the number of players in the system is large).
- A higher μ likely results in less competition, however it also results in a higher probability of player i departing from the system and hence losing out on the reward for the time it stays out; this leads to a tradeoff. Figure 2(b) shows that the effect of the probability of player i departing from the system dominates the effect of the reduction in competition. For similar reasons as above, the expected MPE utility when the number of players in the system is large, can exceed the corresponding SNE utility.

6 FUTURE WORK

One could study a variant of Scenario 1 where the rate of problem getting solved (and perhaps also the cost) increases non-linearly with the invested power. Since players are seldom completely rational in real world, it would be useful to study the game under bounded rationality. To develop a more sophisticated stochastic model, one could obtain real data concerning the arrivals and departures of players and their investment strategies. Another promising possibility is to incorporate state-learning in our model. A Stackelberg game could be studied, where the system decides the amount of reward to offer, and then the computational providers decide how much power to invest based on the offered reward.

APPENDIX

We take cues from bitcoin mining for our numerical simulations. The current offered reward for successfully mining a block is 12.5 bitcoins. Assuming 1 bitcoin \approx \$8000, the reward translates to \$10⁵. The bitcoin problem complexity is set such that it takes around 10 minutes on average for a block to get mined. That is, the rate of problem getting solved is 0.1 per minute on average. One of the most powerful ASIC (application-specific integrated circuit) currently available in market is Antminer S9, which performs computations of upto 13 TeraHashes per sec, while consuming about 1.5 kWh in 1 hour, which translates to \$0.18 per hour (at the rate of \$0.12 per kWh), equivalently \$0.003 per minute. As per BitNode (bitnodes.earn.com), a crawler developed to estimate the size of bitcoin network, the number of bitcoin miners is around 10⁴. Hence, we consider $r = 10^5$, $\gamma = \beta = 0.1$, $c = 0.003$, $|\mathcal{U}| = 10^4$.

ACKNOWLEDGMENT

The work is partly supported by CEFIPRA grant No. IFC/DST-Inria-2016-01/448 “Machine Learning for Network Analytics”.

REFERENCES

- [1] L. F. G. Sarmenta, “Volunteer computing,” Ph.D. dissertation, Massachusetts Institute of Technology, 2001.
- [2] D. P. Anderson and G. Fedak, “The computational and storage potential of volunteer computing,” in *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID’06)*, vol. 1. IEEE, 2006, pp. 73–80.
- [3] Z. Zheng and S. Xie, “Blockchain challenges and opportunities: A survey,” *International Journal of Web and Grid Services*, 2018.
- [4] L. S. Shapley, “Stochastic games,” *Proceedings of the National Academy of Sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.
- [5] E. Maskin and J. Tirole, “Markov perfect equilibrium: I. observable actions,” *Journal of Economic Theory*, vol. 100, no. 2, pp. 191–219, 2001.
- [6] D. Gillette, “Stochastic games with zero stop probabilities,” *Contributions to the Theory of Games*, vol. 3, pp. 179–187, 1957.
- [7] A. M. Fink et al., “Equilibrium in a stochastic n -person game,” *Journal of Science of the Hiroshima University, series A-I (Mathematics)*, vol. 28, no. 1, pp. 89–93, 1964.
- [8] J.-F. Mertens and A. Neyman, “Stochastic games,” *International Journal of Game Theory*, vol. 10, no. 2, pp. 53–66, 1981.
- [9] J. K. Goeree and C. A. Holt, “Stochastic game theory: For playing games, not just for doing theory,” *Proceedings of the National Academy of Sciences*, vol. 96, no. 19, pp. 10 564–10 567, 1999.
- [10] E. Altman, T. Boulogne, R. El-Azouzi, T. Jiménez, and L. Wynter, “A survey on networking games in telecommunications,” *Computers & Operations Research*, vol. 33, no. 2, pp. 286–311, 2006.
- [11] E. Altman, R. El-Azouzi, and T. Jiménez, “Slotted Aloha as a stochastic game with partial information,” in *WiOpt’03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003, p. 9 pages.
- [12] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, “An anti-jamming stochastic game for cognitive radio networks,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 877–889, 2011.
- [13] F. Fu and U. C. Kozat, “Stochastic game for wireless network virtualization,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 84–97, 2013.
- [14] E. Altman, “Non zero-sum stochastic games in admission, service and routing control in queueing systems,” *Queueing Systems*, vol. 23, no. 1–4, pp. 259–279, 1996.
- [15] M. Bowling and M. Veloso, “An analysis of stochastic game theory for multiagent reinforcement learning,” Carnegie-Mellon University Pittsburgh Pennsylvania School of Computer Science Technical Report No. CMU-CS-00-165, Tech. Rep., 2000.
- [16] N. Bellomo, *Modeling complex living systems: A kinetic theory and stochastic game approach*. Springer Science & Business Media, 2008.
- [17] E. Altman and N. Shimkin, “Individual equilibrium and learning in processor sharing systems,” *Operations Research*, vol. 46, no. 6, pp. 776–784, 1998.
- [18] A. Nahir, A. Orda, and D. Raz, “Workload factoring with the cloud: A game-theoretic perspective,” in *IEEE International Conference on Computer Communications (INFOCOM)*, vol. 12. IEEE, 2012, pp. 2566–2570.
- [19] R. Hassin and M. Haviv, “Nash equilibrium and subgame perfection in observable queues,” *Annals of Operations Research*, vol. 113, no. 1–4, pp. 15–26, 2002.
- [20] J. Wang and F. Zhang, “Strategic joining in M/M/1 retrial queues,” *European Journal of Operational Research*, vol. 230, no. 1, pp. 76–87, 2013.
- [21] J. Hu and M. P. Wellman, “Nash Q-learning for general-sum stochastic games,” *Journal of Machine Learning Research*, vol. 4, no. Nov, pp. 1039–1069, 2003.
- [22] C. Jiang, Y. Chen, Y.-H. Yang, C.-Y. Wang, and K. R. Liu, “Dynamic Chinese restaurant game: Theory and application to cognitive radio networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 1960–1973, 2014.
- [23] C.-Y. Wang, Y. Chen, and K. R. Liu, “Game-theoretic cross social media analytic: How Yelp ratings affect deal selection on Groupon?” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 908–921, 2018.
- [24] I. Abraham, D. Dolev, R. Gonen, and J. Halpern, “Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation,” in *ACM Symposium on Principles of Distributed Computing*. ACM, 2006, pp. 53–62.
- [25] Y.-K. Kwok, S. Song, and K. Hwang, “Selfish grid computing: Game-theoretic modeling and NAS performance results,” in *IEEE International Symposium on Cluster Computing and the Grid*. IEEE, 2005.
- [26] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, “A game-theoretic method of fair resource allocation for cloud computing services,” *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252–269, 2010.
- [27] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz, “Selfish caching in distributed systems: A game-theoretic analysis,” in *ACM Symposium on Principles of Distributed Computing*. ACM, 2004, pp. 21–30.
- [28] D. Grosu and A. T. Chronopoulos, “Noncooperative load balancing in distributed systems,” *Journal of Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1022–1034, 2005.
- [29] A. Sapirshstein, Y. Sompolinsky, and A. Zohar, “Optimal selfish mining strategies in bitcoin,” in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 515–532.
- [30] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosen-schein, “Bitcoin mining pools: A cooperative game theoretic analysis,” in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 2015, pp. 919–927.
- [31] Z. Xiong, S. Feng, D. Niyato, P. Wang, and Z. Han, “Optimal pricing-based edge computing resource management in mobile blockchain,” in *IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.
- [32] E. Altman, A. Reiffers, D. S. Menasché, M. Datar, S. Dhamal, and C. Touati, “Mining competition in a multi-cryptocurrency ecosystem at the network edge: A congestion game approach,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 3, pp. 114–117, 2019.
- [33] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, “Blockchain mining games,” in *ACM Conference on Economics and Computation (EC)*. ACM, 2016, pp. 365–382.
- [34] J. H. Hubbard and B. B. Hubbard, *Vector calculus, linear algebra, and differential forms: A unified approach*. Matrix Editions, 2015.
- [35] D. P. Bertsekas and J. N. Tsitsiklis, “Neuro-dynamic programming: an overview,” in *IEEE Conference on Decision and Control (CDC)*, vol. 1. IEEE, 1995, pp. 560–564.